

如何使用AI Code



需要申请AI账号（密钥）的同事，需要先走审批流程，+1和部门负责人通过后才能使用。审批流程可以进入飞书审批->信息资源申请->ai资源申请：链接：

<https://applink.feishu.cn/T94ds4FBF0V4>



如何使用Codex/Gemini/OpenCode

本文仅针对Claude Code，如需使用Codex/OpenCode/Gemini等其他工具可按如下步骤进行：

1. 在服务器安装对应工具，安装CLI的方法见各个工具的官网
2. 登录【紫荆AI Code统筹平台】<http://cc.siorigin.com> 创建对应的Key：
<http://cc.siorigin.com/keys>
3. 在“操作-使用密钥”栏可以看到对应的密钥配置方法

Codex CLI简易安装\使用

安装

代码块

```
1 bash /share_data/codex_install.sh sk-xxx
2 source # source一下，当前窗口才可用
```

验证：

代码块

```
1 codex -V
```

如果提示 `INVALID_API_KEY`，重新运行安装命令写入自己的 Key。

常用启动

代码块

```
1 # 在当前目录启动 Codex
2 codex
3
4 # 在指定项目目录启动
5 codex -C /path/to/project
6
7 # 临时指定模型
8 codex -m gpt-5.5
9
10 # 启用网页搜索
11 codex --search
12
13 # 恢复最近一次会话
14 codex resume --last
```

默认 YOLO 模式

临时使用:

代码块

```
1 codex --dangerously-bypass-approvals-and-sandbox
```

默认开启, 写入 `~/.bashrc`:

代码块

```
1 echo "alias codex='command codex --dangerously-bypass-approvals-and-sandbox'"
  >> ~/.bashrc
2 source ~/.bashrc
```

临时绕过 alias:

代码块

```
1 command codex
```

YOLO 会跳过审批和沙箱, 只建议在隔离好的服务器环境使用。

非交互

代码块

```
1 # 非 git 目录中执行一次性任务
2 codex exec --skip-git-repo-check "总结当前目录"
3
4 # 指定目录和模型执行任务
5 codex exec -C /path/to/project -m gpt-5.5 "找出主要风险"
6
7 # 允许 Codex 修改工作区文件
8 codex exec --sandbox workspace-write "修复测试失败"
9
10 # 把命令输出交给 Codex 分析
11 npm test 2>&1 | codex exec "总结失败原因"
```

类型	用途	常用方式
MCP	接外部工具和服务	codex mcp ...
Skills	固化流程、规范、提示词	\$skill-installer / \$skill-creator
Plugins	插件市场和扩展包	codex plugin marketplace ...
Smithery	找 MCP / Skills	https:// smithery.ai/ skills

MCP 常用：

代码块

```
1 # 查看已配置的 MCP
2 codex mcp list
3
4 # 添加 studio MCP
5 codex mcp add context7 -- npx -y @upstash/context7-mcp
6
7 # 添加 HTTP MCP
8 codex mcp add docs --url https://example.com/mcp
9
10 # 删除 MCP
11 codex mcp remove <name>
```

Skills 常用：

代码块

```
1 # 列出可安装的 skills
2 $skill-installer 列出可安装的 skills
3
4 # 安装某个 skill, 例如 linear
5 $skill-installer linear
6
7 # 创建自己的 skill
8 $skill-creator 帮我创建一个用于 xxx 的 skill
```

Plugins 常用:

代码块

```
1 # 添加插件市场
2 codex plugin marketplace add owner/repo
3
4 # 更新插件市场
5 codex plugin marketplace upgrade
6
7 # 删除插件市场
8 codex plugin marketplace remove <marketplace-name>
```

Claude Code CLI简易安装指南

直接在服务器上安装

1. 安装node和npm

在北京服务器:

代码块

```
1 bash /share/claude_code_env/env-install-wg.sh
```

在上海服务器:

代码块

```
1 bash /share_data/claude_code_env/env-install-wg.sh
```

安装完后记得source ~/.bashrc

无需sudo，它会自己切换到nvm安装npm。可以用which npm检查一下：

代码块

```
1 //npm注意不要用系统里的，可以which npm看一下，确保是在自己目录下，如：  
2 /data/users/shengxiao/.nvm/venprSIONS/node/v24.12.0/bin/npm
```

2. 安装claude-code

代码块

```
1 npm install -g @anthropic-ai/claude-code  
2 //中间可能会要求一个jq的小工具，需要apt install jq，可以找IT sudo安装一下或者用conda安  
装到自己环境里
```

3. 配置claude-code

在北京服务器：

代码块

```
1 bash /share/claude_code_env/env-deploy-wg.sh cr_XXXXXXX
```

在上海服务器：

代码块

```
1 bash /share_data/claude_code_env/env-deploy-wg.sh cr_XXXXXXX
```

其中cr_XXXXXXX为你的API密钥（注意API密钥前缀不一定是cr_），需要申请AI账号（密钥）的同事，需要先走审批流程，+1和最高部门负责人-1通过后才能使用。审批流程可以进入飞书审批->信息资源申请->ai资源申请：链接：<https://aplink.feishu.cn/T94ds4FBF0V4>

!! 重要提醒：

- 请在环境变量里(或~/.bashrc里)增加 `export CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC=1`

这样可以告诉claude code禁止发送任何不必要的网络通信。

- 比如BUG汇报，自动更新之类可用 `export DISABLE_AUTOUPDATER=1` 禁止自动更新。

4. 开始使用：

代码块

- 1 在你的项目下输入claude
- 2 切换模型：输入/model，切换到opus，最强
- 3 切换bypass permission: shift+tab

网上有很多CC使用技巧教程，如：<https://www.youtube.com/watch?v=e5O8A5pcVgg>



更详细使用指南见贾夕阳调研文档 [Claude Code 基础功能与使用方法](#)，包括MCP、Plugin 以及Claude Code最新出的Skill等

5. VS Code / Cursor 插件



- 忒忒，青葵反馈插件编程效果没有 CLI 好
- 插件和 CLI 的历史没有同步

1. 文档：<https://code.claude.com/docs/en/vs-code>

2. 插件：<https://open-vsx.org/extension/Anthropic/claude-code>

Search sessions...

Today

Triton compiler extension for SIPU hardware support 25m

Untitled 54m

你是谁 59m

No prompt 1h

<local-command-stdout>Bye! </local-command-std... 1h

User Exits CLI Session 1h

<local-command-stdout>Bye! </local-command-std... 1h

Claude Code Session Ended 1h

怎样使用 claude code 2h

使用方法、

基本操作

启动：在终端运行 `claude` 命令即可启动 常用快捷键：

- `Ctrl+C` - 取消当前操作或退出
- `Ctrl+L` - 清空当前输入
- `↑/↓` - 浏览历史记录
- `Tab` - 自动补全

常用斜杠命令

命令	功能
<code>/help</code>	显示帮助信息
<code>/clear</code>	清空对话历史
<code>/init</code>	初始化 CLAUDE.md 项目文档
<code>/review</code>	审查代码更改
<code>/compact</code>	压缩对话上下文

主要功能

1. 代码理解 - 自动读取和分析项目代码
2. 代码编辑 - 直接编辑文件、修复 bug
3. 命令执行 - 运行 shell 命令、构建、测试
4. Git 集成 - 创建 commit、PR、查看 diff
5. 多文件操作 - 跨文件搜索、重构

配置文件

- `~/.claude/settings.json` - 全局设置
- `.claude/` 目录 - 项目级配置
- `CLAUDE.md` - 项目上下文文档

你当前的环境

你正在 `siorigin_triton` 项目中，分支 `support-tile-move-fp8`。可用的自定义技能：

- `/review` - PR 审查
- `/pr-comments` - 获取 PR 评论
- `/security-review` - 安全审查

使用技巧



3. 快捷键:

- a. Alt + K: @选中的内容, 类似cursor Ctrl + L

FAQ

1. 如果在自己笔记本电脑上配置, 参考如下流程:

a. Windows:

 **安装步骤**  [有疑问? 技术支持](#)

- 1 【此步骤在桌面】下载git**
访问 <https://git-scm.com/downloads/win>, 安装时全都下一步, 不要修改路径
- 2 【此步骤在桌面】下载nodejs**
访问 <https://nodejs.org/zh-cn/download>, 安装时全都下一步, 不要修改路径
- 3 【此步骤在Windows PowerShell】验证安装情况**
打开Windows PowerShell (蓝色图标), 执行以下命令验证安装情况:

```
PS> node -v  
PS> npm -v
```

若提示「No suitable shell found」, 是git没装好。请将 `CLAUDE_CODE_GIT_BASH_PATH=C:\Program Files\git\bin\bash.exe` 设置到系统环境变量中, 重启终端再试。依旧无效, 重装git, 重启终端后再试
- 4 【此步骤在Windows PowerShell】卸载已安装的Claude Code (未安装请跳过)**

```
PS> npm uninstall -g @anthropic-ai/claude-code
```
- 5 【此步骤在Windows PowerShell】安装官方原包**

```
PS> npm install -g @anthropic-ai/claude-code
```
- 6 【此步骤在控制面板】设置Windows系统环境变量**
需要设置以下三个环境变量:
变量名: ANTHROPIC_BASE_URL, 变量值: 
变量名: ANTHROPIC_API_KEY, 变量值: 你的密钥
变量名: ANTHROPIC_AUTH_TOKEN, 变量值: 你的密钥
设置方法见上文「预备知识: 修改环境变量」
- 7 【此步骤在Windows PowerShell】重启Windows PowerShell, 验证安装结果**
重启Windows PowerShell后运行以下命令, 确认安装成功

```
PS> claude -v
```

从平台上的密钥旁边的使用密钥获取

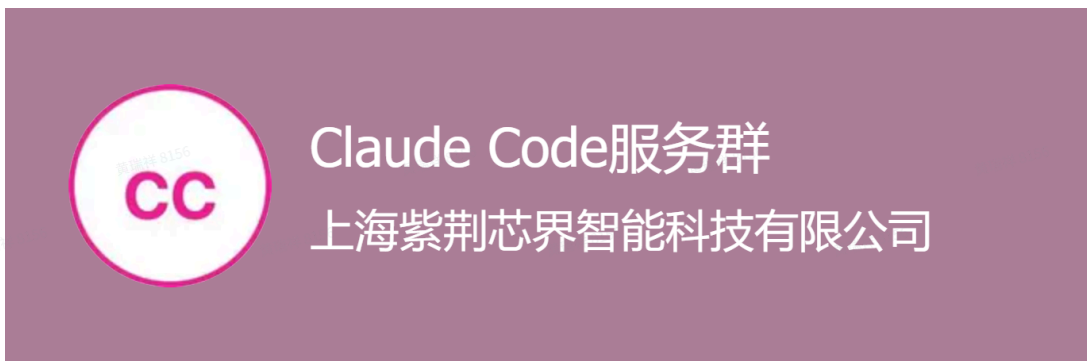
代码块

```
1 ANTHROPIC_BASE_URL="http://cc.siorigin.com/api"  
2 ANTHROPIC_API_KEY="cr_XXXXXXXXX"  
3 ANTHROPIC_AUTH_TOKEN=""
```

2. 如果使用时遇到time out超时，可能是DNS解析问题，可以修改~/.bashrc中的ANTHROPIC_BASE_URL中cc.siorigin.com为10.96.11.131:3100解决，或者联系IT解决该服务器的DNS问题。

```
# Claude Code Environment Variables
export ANTHROPIC_BASE_URL="http://10.96.11.131:3100/api"
```

3. From 周子易：claude经常会尝试在tmp/claude路径下创建临时文件，然后就因为权限问题一直报错Error: EACCES: permission denied, mkdir '/tmp/claude/...'。尝试设置过TMPDIR环境也没有用。试图让claude自己修也修不好
 - a. [BUG] Background tasks ignore \$TMPDIR and hardcode /tmp/claude/ · Issue #15700 · anthropics/claude-code 提到目前的workaround是配置CLAUDE_CODE_TMPDIR这个环境变量，试了一下确实可以，不过还是改权限最方便，大家就不用手动配了
4. 如有其他问题欢迎进群讨论：



仅限企业内部成员加入

该二维码永久有效

MCP使用参考

Claude Code 基础功能与使用方法

JIRA集成

生成 token



使用

代码块

```
1 # 设置环境变量为生成的token
2 export JIRA_TOKEN="XXXXXX"
3
4 # 测试生成token
5 curl -k -H "Authorization: Bearer $JIRA_TOKEN"
  https://jiraoffice.siorigin.com/rest/api/2/myself
```

Claude Code 和 Cursor 配置

1. `pip install mcp-atlassian`
2. 添加配置到文件 `~/.cursor/mcp.json` for cursor, `~/.claude.json` for claude code; 替换 `JIRA_PERSONAL_TOKEN`为用户自己的token

代码块

```
1
2 {
3   "mcpServers": {
4     "mcp-atlassian": {
5       "command": "/opt/conda/bin/mcp-atlassian",
6       "args": ["--no-jira-ssl-verify"],
7       "env": {
8         "JIRA_URL": "https://jiraoffice.siorigin.com",
9         "JIRA_PERSONAL_TOKEN": "XXXXXXXXXXXXXXXXXXXXXXXXX"
10      }
11    }
12  }
```

```
12 }  
13 }
```

Codex设置

~/codex/config.toml

代码块

```
1 [mcp_servers.mcp-atlassian]  
2 command = "/opt/conda/bin/mcp-atlassian"  
3 args = ["--no-jira-ssl-verify"]  
4  
5 [mcp_servers.mcp-atlassian.env]  
6 JIRA_URL = "https://jiraoffice.siorigin.com"  
7 JIRA_PERSONAL_TOKEN = "XXXXXXXXXXXXXXXXXXXXXXX"
```

查看我有哪些Jira要处理

查看我有哪些Jira要处理

mcp_mcp-atlassian_jira_search tool input (otyfe2)

mcp_mcp-atlassian_jira_search

```
IN {
  "jq1": "assignee = currentUser() AND status not in (Closed, Done, Resolved) ORDER BY priority DESC, updated DESC",
  "limit": 20
}
```

```
OUT {"result": {"total": 12, "start_at": 0, "max_results": 20, "issues": [{"id": "10463", "key": "S1SW-7
```

你有 12 个分配给你的 Jira 任务，按优先级和更新时间排序：

处理中 (2个)

Key	标题	优先级	更新时间
S1SW-7	【Triton】支持Siorigin-Triton的Single Tile Matmul算子	Medium	2026-01-06
S1SW-559	【Triton】tl.dot问题	Medium	2025-07-09

待办 (8个)

Key	标题	优先级	更新时间
S1SW-1744	[TRITON] EMU torch sipu AI fill triton kernel 运行很慢	Medium	2025-12-31
S1SW-648	【Triton】tl.log问题	Medium	2025-10-15
S1SW-701	【Triton】tl.where问题	Medium	2025-10-15
S1SW-1119	【TRITON】index_put支持多线程情况	Medium	2025-10-15
S1SW-1129	[TRITON] tensormap dump功能影响大batch bmm结果	Medium	2025-10-15
S1SW-574	[TRITON] siorigin-triton 适配 triton v3.3.1 nvm intrinsic	Medium	2025-07-01
S1SW-288	【Triton】验证 siorigin-triton代码是否支持 triton 官方 repo 最新修改的带 TMA flash attention tutorial	Medium	2025-04-17
S1SW-51	【Triton】支持Siorigin-Triton的tile strided store	Medium	2025-04-17

其他 (2个)

Key	标题	状态	优先级
S1SW-1150	混合编译函数支持: std::bit_cast	Won't Fix	Medium
S1SW-19	【Triton】支持Triton的matmul算子Lowering到RV Core。	待办	Low

最新更新是 [S1SW-7](#) (Single Tile Matmul算子)，今天刚更新过。

S1SW 项目 - Triton 相关 Jira 状态

总体统计

类别	数量
未解决	48
已完成	83
Won't Fix	2
总计	248

未解决 Issues 状态分布

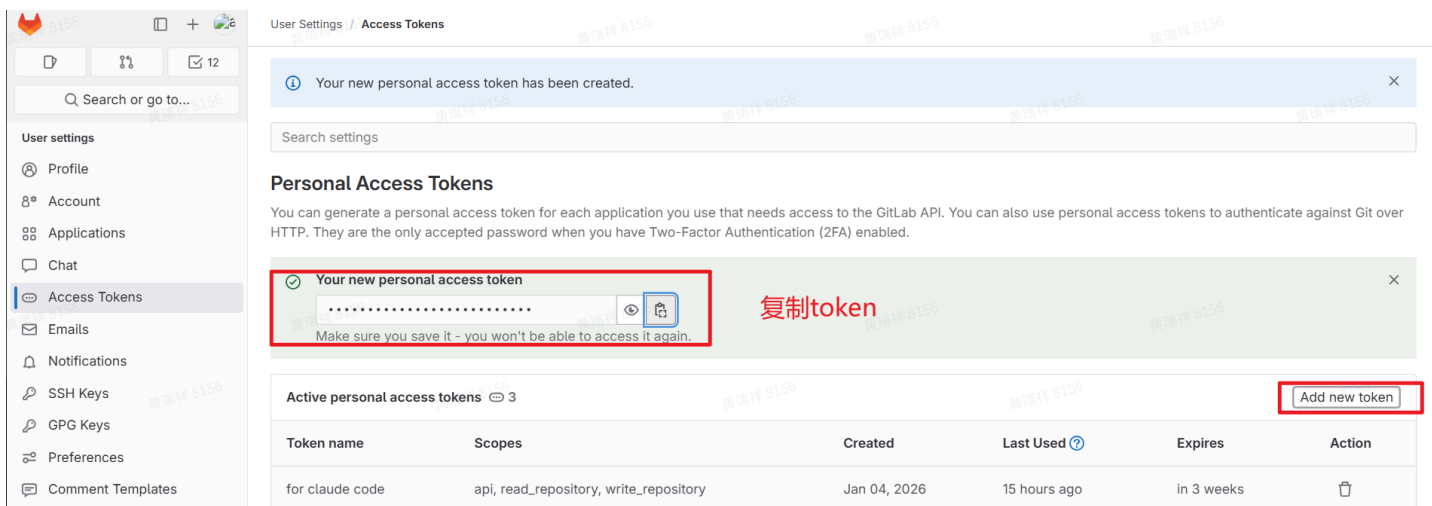
状态	数量
● 处理中	10
● 待办	20
● 开放	12
II Suspend/Pending	5
✔ Verify	1

● 处理中 (10个)

Issue	优先级	负责人	标题
S1SW-1363	Medium	解义	在Triton.autotune的config阶段, 对齐sipu的参数配置
S1SW-1287	Medium	刘东辉	triton 支持 atomic_add 处理 tensor 以及 mask
S1SW-1263	Medium	解义	实现针对torch.compile生成的triton kernel自动对数功能
S1SW-1213	Medium	解义	【Triton】【Torch Compile】Autotune/Heuristic开发汇总
S1SW-709	Medium	刘东辉	【Triton】 tl.range问题
S1SW-624	Medium	刘东辉	【Triton】 tl.sum问题
S1SW-580	Medium	冯国庆	【Triton】 tl.broadcast问题
S1SW-578	Medium	冯国庆	【Triton】 tl.max问题
S1SW-559	Medium	贾夕阳	【Triton】 tl.dot问题
S1SW-7	Medium	贾夕阳	【Triton】 支持Siorigin-Triton的Single Tile Matmul算子

gitlab集成

生成 token



Claude Code 和 Cursor 配置

1. 添加配置到文件 `~/.cursor/mcp.json` for cursor, `~/.claude.json` for claude code; 替换 `GITLAB_PERSONAL_ACCESS_TOKEN`为用户自己的token

代码块

```

1  {
2    "mcpServers": {
3      "gitlab": {
4        "command": "npx",
5        "args": [
6          "-y",
7          "@zereight/mcp-gitlab"
8        ],
9        "env": {
10         "GITLAB_PERSONAL_ACCESS_TOKEN": "XXXXXXXXXXXX",
11         "GITLAB_API_URL": "https://gitlabsoft.siorigin.com"
12       }
13     }
14   }
15 }

```

Codex设置

`~/.codex/config.toml`

代码块

```

1  [mcp_servers.gitlab]
2  command = "npx"
3  args = ["-y", "@zereight/mcp-gitlab"]
4
5  [mcp_servers.gitlab.env]
6  GITLAB_PERSONAL_ACCESS_TOKEN = "XXXXXXXXXXXXXXXXXXXX"

```

7 GITLAB_API_URL = "https://gitlabsoft.siorigin.com"

查看我最近提交的MR

no_bug_car.txt

- **mcp_gitlab_list_merge_requests**

```
IN {
  "author_username": "jiaxiyang",
  "order_by": "created_at",
```

```
OUT []
```

- **mcp_gitlab_list_merge_requests**

```
IN {
  "order_by": "created_at",
  "sort": "desc",
```

```
OUT [
  {
    "id": "8272",
```

- 你最近提交的 MR 如下:

最近 10 个 MR (siorigin_triton)

状态	MR	标题	创建时间
✓ merged	!371	[DOCS] Update commit-push-pr command with MCP tools and squash settings	2026-01-07
✗ closed	!370	[DOCS] Update commit-push-pr command to use MCP GitLab tools	2026-01-07
✓ merged	!368	[FEAT] Add Claude command for creating Jira issues with MCP validation	2026-01-06
✓ merged	!367	[BUILD] Update SIPU SDK version to 260106	2026-01-06
✓ merged	!365	[BUILD] Install torch from local wheel in Dockerfile	2026-01-06
✓ merged	!363	[DOCS] Enhance commit-push-pr workflow with glab integration	2026-01-05
✓ merged	!362	[DOCS] Disable AI attribution in Claude Code settings	2026-01-05
✓ merged	!358	[DOCS] Add Claude Code configuration and skills	2026-01-05
✓ merged	!355	[DOCS] Add Claude Code configuration and project documentation	2026-01-04
✓ merged	!350	[FIX] Improve RVV rsqrt precision with Newton-Raphson iteration	2025-12-31

统计: 10 个 MR, 9 个已合并, 1 个已关闭

总结一下最近的提交做了什么

claude-code.md

- 根据最近的 MR 记录, 你最近的工作主要集中在以下几个方面:

文档与工具配置 (文档工程)

- **MCP 工具集成:** 将 GitLab MCP 工具集成到 commit-push-pr workflows 中, 替代了 glab CLI
- **Claude Code 配置:** 添加了技能、权限设置、项目文档 (CLAUDE.md)
- **Jira 集成:** 添加了通过 MCP 创建 Jira issue 的命令

构建相关

- **SDK 版本更新:** SIPU SDK 从 251212 升级到 260106

- Dockerfile 优化: 改为从本地 wheel 文件安装 torch

功能开发

- RVV rsqrt 精度优化 (S1SW-1706): 使用 Newton-Raphson 迭代提高精度到 ~23 bits